Methods in Computational Linguistics I

Course Number: LING78100

Thursdays, 11:45am – 1:45pm, Room TBA

Four classes will be held in person in Graduate Center:

- Aug. 26
- Sept. 2
- Oct. 21
- Nov. 18

The remaining eleven class sessions will be held using the Blackboard Ultra remote teaching capability.

Instructor Information:

Elena Filatova, PhD

CST Department at NYCCT

CUNY Grad Center CS, Linguistics programs

Email: efilatova@citytech.cuny.edu

Web: https://ef2020.commons.gc.cuny.edu/

Office hours: By appointment (request via email)

Fall 2021 LING 78100 (including this syllabus) is based on LING 78100 previously taught by Prof. K. Gorman, Prof.M. Mandel, Prof. A. Rosenberg and Rachel Rakov.

Practicum Sessions and Instructor Information:

This course has a co-requisite practicum session in which you should also enroll: Ling73600. The practicum will meet in room TBA on Fridays from 11:45pm to 1:45pm. The instructor for the practicum is Arundhati Segupta, Ph.D. student in Linguistics, Graduate Center. During the practicum session, you will review concepts from class and see extended programming examples.

Description:

This is the first of a two-part course sequence to train students with a linguistics background in the core methodologies of computational linguistics. Successful completion of this two-course sequence will enable students to take graduate-level elective courses in computational linguistics. Both courses offered by the Graduate Center's Linguistics program, as well as courses offered by the Computer Science program. As the first part of the two-part sequence, Methods in Computational Linguistics I will introduce computer programming at a level that will allow students to begin building computer applications that address various computational linguistic tasks. No previous programming experience is required. The programming language we will use is Python. We begin by learning the syntax of Python and how to program generally; we then focus specifically on linguistic application. Who Should Take This Course: This course is required for students pursuing the MA in Computational Linguistics or the PhD Certificate in Computational Linguistics at CUNY Graduate Center. Further, this course would be excellent for students who may be interested in research in computational linguistics or natural language processing (NLP). Other graduate students (including those outside of linguistics) who wish to gain basic programming skills in the Python language, which is useful for text processing and various linguistics and web applications, may also benefit from this course. Because this course introduces basic programming concepts, it would not be appropriate for graduate students in Computer Science.

Online:

Lecture slides, assignments, and handouts will be available on the course CUNY Blackboard.

Assignments should be submitted to the relevant dropbox on the course's page on Blackboard.

A Note on Pre-requisites:

There are no pre-requisites for this course. However, this course is a prerequisite for Methods in Computational Linguistics II, usually offered in the spring.

Required Books:

Books 1 – 4 are free on-line books. Following the course pace there might be additional reading materials posted on the class blackboard.

1. Think Python (2nd Edition) by Allen B. Downey: http://greenteapress.com/wp/think-python-2e/. Following this link you will find:
    - Downloadable PDF version of the book (http://greenteapress.com/thinkpython2/thinkpython2.pdf)
    - HTML (on-line) version of the book (http://greenteapress.com/thinkpython2/html/index.html)
    - Example programs and solutions to some exercises
2. How to Think Like a Computer Scientist by Peter Wentworth, Jeffry Elkner, Allen B. Downey, and Chris Meyers: http://www.openbookproject.net/thinkcs/python/english3e/
3. Bird, S., Klein, E. and Loper, E. n.d. Natural language processing with Python. URL: https://www.nltk.org/book/.
4. Church, K. W. No date. Unix™ for poets. URL: http://doc.cat-v.org/unix/for-poets/kwc-unix-for-poets.pdf.

5. Jurafsky, D., and Martin, J. H. 2019. Speech and language processing. 3nd edition. Pearson. (See on-line notes: https://web.stanford.edu/~jurafsky/slp3/.)

Technical Requirements:

For assignments in this course, students will be expected to write computer programs in the Python programming language. Students may also be expected to take advantage of resources in the Natural Language Toolkit (NLTK) programming library. Students are expected to install Python 3.6 and the Natural Language Toolkit (NLTK) on their personal computers. Details about what specific versions of this software to install and how to configure it will be discussed during the first week of class. Students are advised to bring laptop computers to meetings of the practicum sessions.

Learning Goals:

By the end of the semester, students will:

1. Understand the basics of computer programming, including: names, functions, numerical and string data types, list-like container data types, control of flow, and other key concepts.
2. Be able to write short programs in the Python programming language.
3. Be familiar with the use of software libraries for computational linguistic programming.
4. Be able to accomplish some computer programming tasks that are useful for working in the field of computational linguistics, including: processing a text file or other basic string operations.
5. Be familiar with the basic topics that make up the field of computational linguistics.
6. Be prepared to learn more advanced programming and mathematical topics in the subsequent "Methods in Computational Linguistics II" course, which is the second half of this two-course sequence.

Outline of Course Topics:

- Computer Basics, Programming Code
- Introduction to Python and the IDLE editor vs Jupyter Notebooks
- Processing with Numbers: Data Types and Conversions
- String Processing with Python
- Opening, Reading, and Writing Data Files
- Booleans and Conditionals
- Basic Graphics and User-Interface Issues
- Designing Functions, Top-down Software Design
- Loops, Control of Flow, List Comprehensions
- Python Containers: Lists, Tuples, Dictionaries
- Defining and Using Classes, Object-Oriented Software Design
- Overview of Topics in Computational Linguistics
- Introduction to Machine Learning

Assignments and Grading:

Details regarding due-dates for various homework assignments will be listed on the course blackboard, as well as announced in class.

Only electronic homework assignment submission via the class blackboard will be accepted. No late submission will be accepted.

Attending class is mandatory and attendance will be taken at the beginning of every meeting. If there is an important reason or circumstances beyond your control why you need to be late or miss the class, write me an email before the beginning of the class.

Grading Percentages for the Course:

- Regular homework assignments 80%
- Final homework assignment 10%
- Class participation & attendance 10%

Assignment Policy:

Do not cheat. Unless announced otherwise, all the assignments are individual assignments. You may discuss assignments with your classmates, course instructor, or the practicum instructor, but the program that you hand in must be your own. Do not ask for or offer to share code or written assignments. If you discuss an assignment with a classmate or the practicum leader, you must indicate this in the documentation of your code, along with the name of the classmate or practicum leader. The first instance of cheating results in an automatic zero for the assignment (or final project). The second instance of cheating results in a zero (F) for the course. The Linguistics Department will be notified in writing of all instances of cheating. On a second instance, a report will be submitted to the Office of Academic Integrity.

Incomplete Policy:

In extenuating circumstances, students may be given an Incomplete if material has not been completed by the end of the semester. When an incomplete is granted, the student and instructor will specify, in writing, a timeframe for all outstanding material to be submitted. If no other timeframe has been specified in writing, the deadline for all outstanding material to be submitted to resolve an incomplete will be one month following the last meeting of the class. An incomplete that is not resolved by the deadline will become an F.

Class Policies:

- Come to class. It will be difficult to do well in the class without regular attendance.
- Cell phones must be on silent, and are not to be checked or used during class - if you are expecting an urgent call, tell the instructor at the start of class.
- Laptops, tablets or lab computers are welcome in class so long as the sound is turned OFF.

Advice for all Students:

- **Begin assignments early!** When you are new to programming, it can be difficult to judge how much time you will need to write a program. Writing the first version of the program is only the start, it can take much longer to find the bugs in it and get it working correctly.

- Review the reading material before class so that you can ask better questions during class.